

EasyAVR[®] 6

Manuel utilisateur

Les systèmes de développement MikroElektronika sont des outils irremplaçables pour le développement et la programmation des microcontrôleurs. Un choix attentif des composants ainsi que l'utilisation d'appareils de dernière génération pour le montage et le test constitue la meilleure garantie de fiabilité de nos produits. Grâce à leur simplicité, leur nombre important de modules intégrés et d'exemples prêt à l'emploi, tous nos utilisateurs, indépendamment de leur expérience, ont la possibilité de réaliser leur projet de manière rapide et efficace.

Systeme de developpement

 **MikroElektronika**

SOFTWARE AND HARDWARE SOLUTIONS FOR EMBEDDED WORLD ...making it simple

CHER CLIENT,

Je tiens à vous remercier pour l'intérêt que vous avez porté à nos produits et pour la confiance que vous avez accordée à MikroElektronika.

Notre objectif est de vous fournir des produits de la meilleure qualité possible. En outre, nous continuerons à améliorer nos performances afin de répondre à vos besoins.



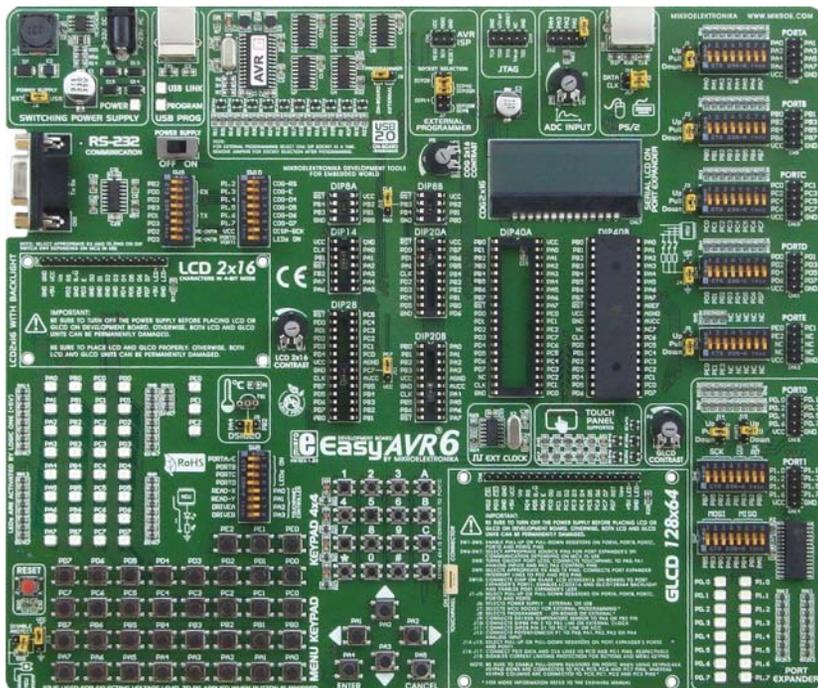
Nebojsa Matic
General Manager

TABLE DES MATIERES

Présentation du système de développement EasyAVR6	4
Caractéristiques principales	5
1.0. Connexion au PC	6
2.0. Microcontrôleurs supportés	7
3.0. Programmateur embarqué AVRprog USB 2.0	8
4.0. Programmateur externe AVRISP mkII	9
5.0. Connecteur JTAG	10
6.0. Oscillateur	10
7.0. Alimentation électrique	11
8.0. Interface de communication RS-232	12
9.0. Interface de communication PS/2	13
10.0. Capteur de température DS1820	14
11.0. Convertisseur Analogique/Numérique (ADC).....	15
12.0. LEDs	16
13.0. Boutons poussoirs	17
14.0. Claviers numériques	18
15.0. Afficheur LCD alphanumérique 2x16	19
16.0. Afficheur LCD 2x16 embarqué	20
17.0. Afficheur graphique LCD 128x64	21
18.0. Panneau tactile (<i>touch panel</i>)	22
19.0. Port d'Entrées/Sorties	23
20.0. Extenseur de port (port E/S additionnels)	25

Présentation de la carte de développement EasyAVR6

Le système de développement *EasyAVR6* est un outil exceptionnel destiné à la programmation et à l'expérimentation sur les microcontrôleurs AVR® développés par *Atmel*®. La carte inclue un programmeur servant d'interface entre le microcontrôleur et le PC. Vous n'avez donc qu'à écrire votre code dans un de nos compilateurs, générer un fichier .hex puis programmer votre microcontrôleur grâce au programmeur *AVRprog*®. De nombreux modules embarqués, comme l'afficheur LCD graphique 128x64, l'afficheur LCD 2x16, l'afficheur LCD embarqué 2x16 LCD, le clavier numérique 4x4, l'extenseur de port etc, vous seront d'une grande aide pour la simulation.



Carte de développement toute option et conviviale pour microcontrôleur AVR



Programmeur haute performance USB 2.0 embarqué



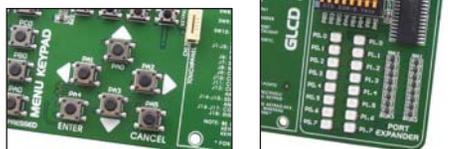
Extenseur de port pour une extension facile d'E/S (2 ports additionnels)



Afficheur alphanumérique LCD 2x16 série embarqué



Afficheur LCD Graphique avec rétro-éclairage



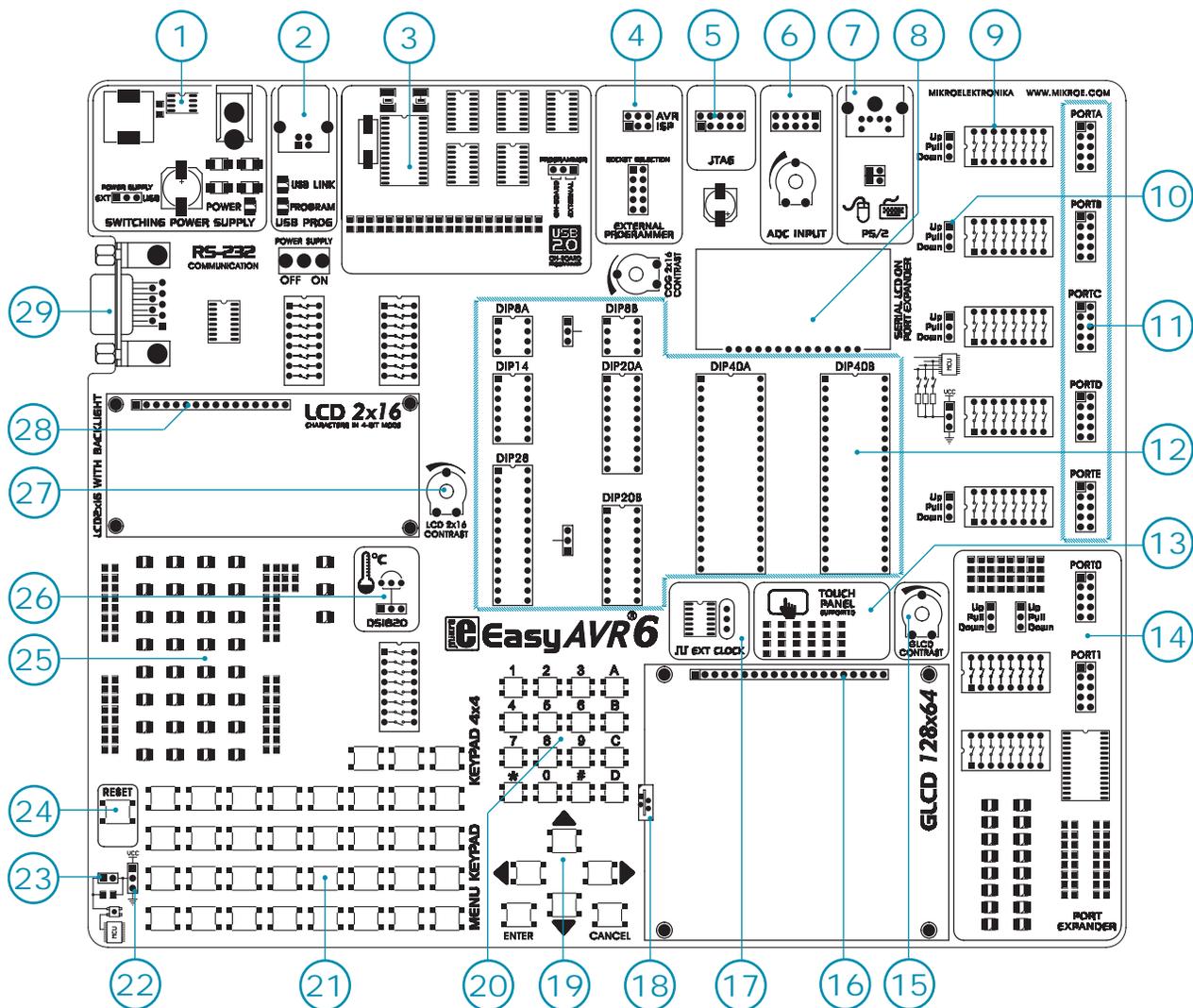
Le programme *AVRflash* fournit la liste complète de tous les microcontrôleurs supportés. La dernière version de ce programme avec la liste mise à jour est téléchargeable à partir de notre site www.mikroe.com

Contenu du pack:

Carte de développement:	<i>EasyAVR6</i>
CD:	CD contenant les logiciels
Câbles:	Câble USB
Documentation:	Manuels <i>EasyAVR6</i> et <i>AVRflash</i> , manuel <i>Installation drivers USB</i> et schéma électronique <i>EasyAVR6</i>

Spécifications du système:

Alimentation:	par connecteur AC/DC (de 7V à 23V AC ou de 9V à 32V DC); par câble USB (5V DC)
Consommation:	50mA en mode <i>idle</i> (lorsque les modules de la carte sont inactifs)
Dimensions:	26,5 x 22cm (10,4 x 8,6inch)
Poids:	~417g (0.92lbs)



Caractéristiques principales

- | | |
|--|---|
| <ol style="list-style-type: none"> 1. Alimentation régulée en tension 2. Connecteur USB du programmeur embarqué 3. Programmeur USB 2.0 AVRprog 4. Connecteur pour programmeur externe AVRISP® 5. Connecteur interface JTAG® 6. Entrées de tests du convertisseur A/N 7. Connecteur PS/2 8. Afficheur LCD 2x16 embarqué 9. DIP Switches pour activer les résistances de tirage 10. Sélection des modes pull-up/pull-down 11. Connecteurs des ports d'E/S 12. Interfaces de connexion pour microcontrôleur AVR 13. Contrôleur du panneau tactile 14. Extenseur de port | <ol style="list-style-type: none"> 15. Contraste de l'afficheur LCD graphique 128x64 16. Connecteur de l'afficheur LCD graphique 128x64 17. Horloge du microcontrôleur 18. Connecteur du panneau tactile 19. Clavier numérique MENU 20. Clavier numérique 4x4 21. Boutons poussoirs pour la simulation des entrées numériques 22. Sélecteur d'état logique 23. Jumper ON/OFF de la résistance de protection 24. Bouton RESET 25. 35 LEDs d'indication de l'état logique des broches d'E/S 26. Connecteur du capteur de température DS1820 27. Contraste de l'afficheur LCD alphanumérique 28. Connecteur de l'afficheur LCD alphanumérique 29. Connecteur pour la communication RS-232 |
|--|---|

1.0. Connexion au PC

Etape 1:

Suivez les instructions pour l'installation des pilotes USB et du programmeur *AVRflash* fournies dans les manuels associés. Il n'est pas possible de programmer les microcontrôleurs AVR sans avoir préalablement installé ces pilotes. Dans le cas où un des compilateurs MikroElektronika est déjà installé sur votre PC, il n'est pas nécessaire de réinstaller *AVRflash* (dans la mesure où celui-ci a été installé en même temps que le compilateur)

Etape 2:

Utilisez le câble USB fourni pour connecter le système de développement *EasyAVR6* à votre PC. Une extrémité du câble USB (fiche USB **B**) devra être connectée à la carte de développement comme décrit Figure 1-2, tandis que l'autre extrémité (fiche USB **A**) devra être connectée à votre PC. Avant d'établir la connexion, assurez vous que le *jumper* J6 est bien placé sur la position USB comme indiquée Figure 1-1.

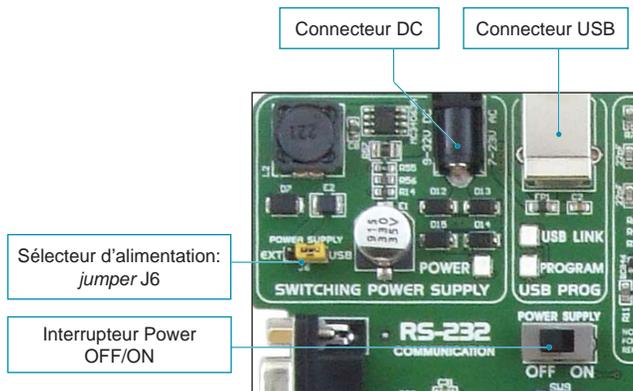


Figure 1-1: Alimentation

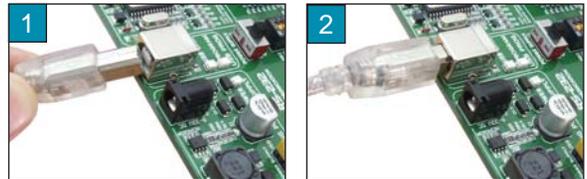


Figure 1-2: Connexion du câble USB (*jumper* J6 en position USB)

Etape 3:

Démarrez votre système de développement en poussant l'interrupteur d'alimentation sur la position ON. Les LEDs POWER et USB LINK vont s'allumer, signifiant que votre système de développement est prêt à être utilisé. Utilisez *AVRflash* pour charger votre code dans le microcontrôleur et servez vous de la carte pour tester et développer votre projet.

NOTE: Si vous souhaitez ajouter des modules complémentaires comme des LCD, GLCD, cartes supplémentaires etc., il est indispensable de les placer correctement avant la mise sous tension. Dans le cas contraire, ils pourraient être définitivement endommagés.

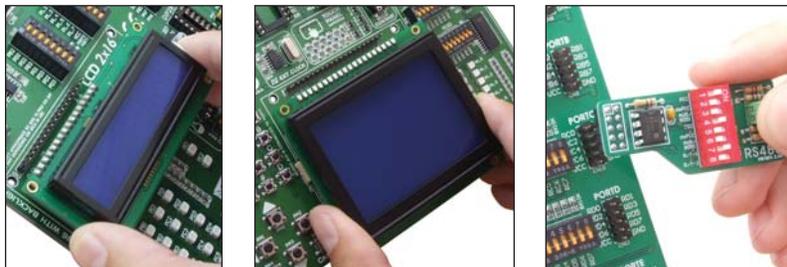


Figure 1-3: Placement des modules complémentaires sur la carte

2.0. Microcontrôleurs supportés

Le système de développement *EasyAVR6* dispose de huit interfaces de connexion différents pour microcontrôleurs (*socket*) AVR: DIP40, DIP28, DIP20, DIP14 et DIP8 . Ces interfaces permettent une connexion direct du microcontrôleur à la carte.

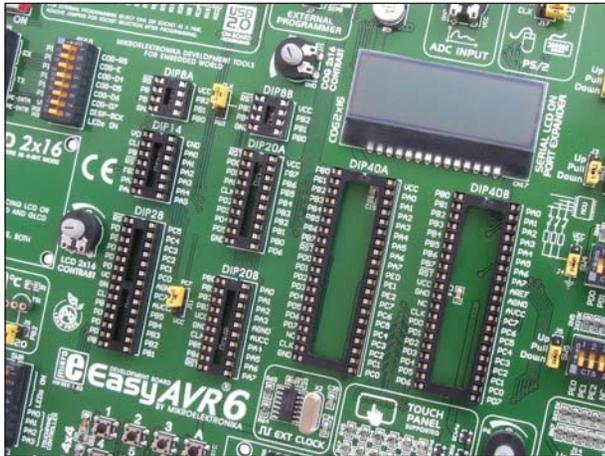


Figure 2-1: Sockets des microcontrôleurs

La carte possède deux *sockets* pour la connexion des AVR de formats DIP40, DIP20 et DIP8. Ces interfaces diffèrent par leur brochage. Le microcontrôleur livré avec *EasyAVR6* est placé sur un interface DIP40.

Les *jumpers* J10 et J11 situés à coté des *sockets* DIP28 et DIP8 servent à préciser la fonction de certaines broches du microcontrôleur:

Jumper	Position	Fonction
J10	PB3	PB3 est une broche E/S
	OSC	La broche PB3 est reliée au signal d'horloge issue de l'oscillateur de la carte
J11	VCC	La broche est connectée à VCC
	PC7	PC7 est une broche E/S

Les microcontrôleurs AVR peuvent utiliser aussi bien leur oscillateur interne que l'oscillateur externe (embarqué sur la carte) comme source de signal d'horloge. L'oscillateur fourni sur la carte génère un signal d'horloge pour la plupart des microcontrôleurs supportés.

- Les microcontrôleurs connectés à DIP8A utilisent leur oscillateur interne (il ne sont pas reliés à l'oscillateur de la carte).
- Les microcontrôleurs connectés à DIP8B peuvent utiliser aussi bien leur oscillateur interne que l'oscillateur externe (position du *jumper* J10).

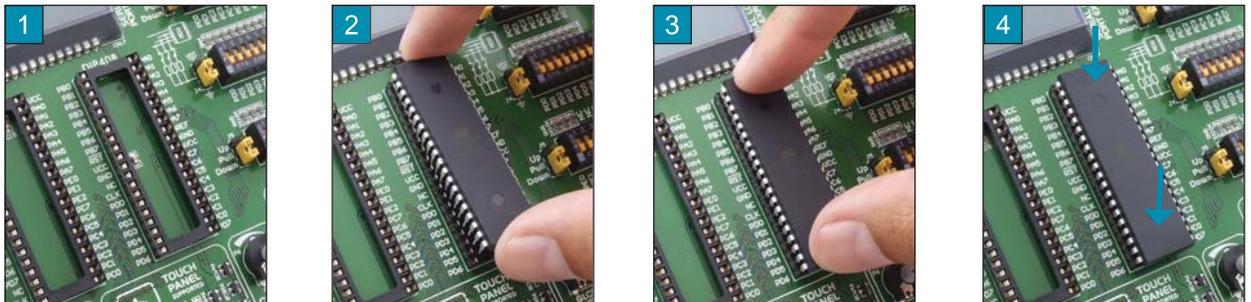


Figure 2-2: Connexion du microcontrôleur à l'interface appropriée

Avant de connecter le microcontrôleur, assurez vous que l'interrupteur de l'alimentation est sur OFF. La Figure 2-2 vous montre comment connecter correctement un microcontrôleur. La Figure 1 montre une interface de connexion DIP40 non utilisée. Placez une extrémité du microcontrôleur sur l'interface, voir Figure 2. Puis enfoncez lentement le microcontrôleur jusqu'à que toutes les broches soient correctement positionnées au dessus de leur emplacement respectif comme décrit Figure 3. Vérifiez que toutes les broches sont parfaitement positionnées et pressez lentement le microcontrôleur jusqu'à ce qu'il soit complètement enfoncé, voir Figure 4.

NOTE: Un seul microcontrôleur à la fois peut être connecté à la carte de développement.

3.0. Programmeur embarqué AVRprog USB 2.0

Le programmeur *AVRprog* est un outil dont le but est de charger un fichier hexadécimal dans le microcontrôleur. Le *EasyAVR6* possède un programmeur embarqué *AVRprog* qui vous permet d'établir la communication entre le microcontrôleur et votre PC. Le contrôle du programmeur s'effectue grâce au programme *AVRflash*. La Figure 3-2 présente les liens entre le compilateur, le programmeur et le microcontrôleur.

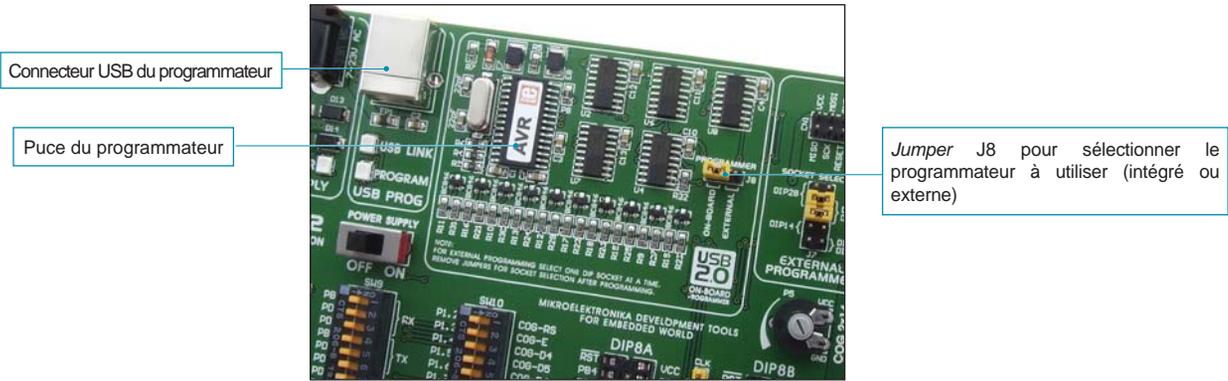


Figure 3-1: Programmeur AVRprog

- ① Ecrivez un programme dans un des compilateurs AVR et générez le fichier HEX;
- ② Dans *AVRflash*, sélectionnez le microcontrôleur approprié et chargez le code HEX dans le *buffer* du programmeur;
- ③ Cliquez sur le bouton *Write* et chargez le programme dans le microcontrôleur.

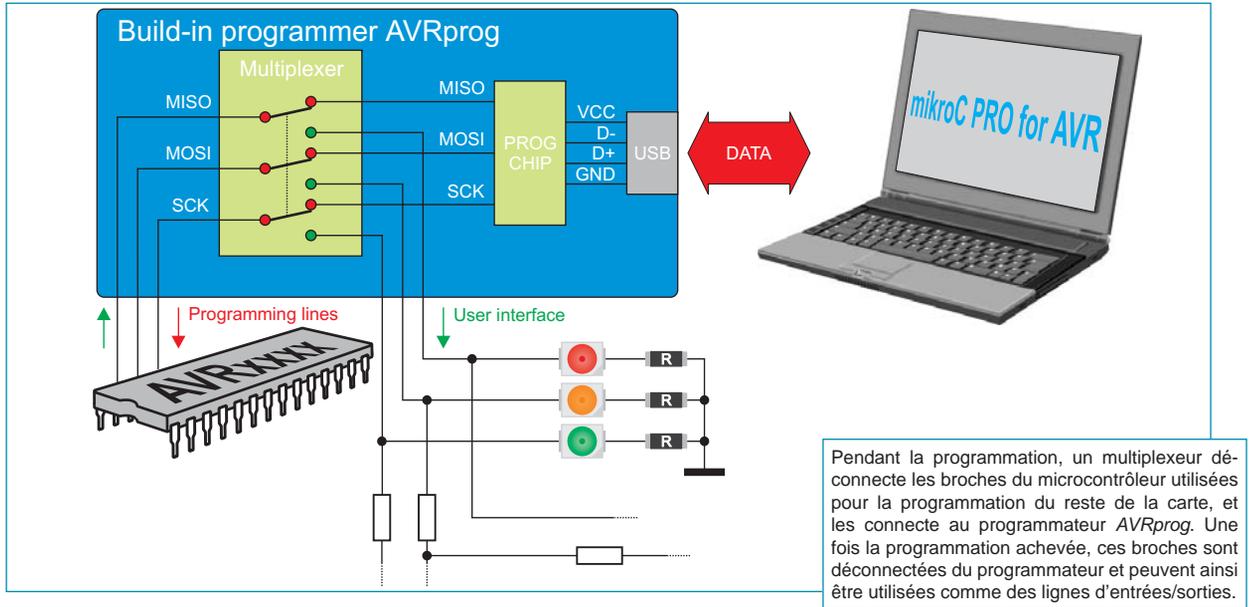
Des boutons facilitant la programmation sont disponibles sur la partie droite. L'option *Progress* située en bas de la fenêtre indique la progression du processus de programmation.

Ecrivez votre code dans un des compilateurs AVR, générez un fichier .hex. Le programmeur se chargera du chargement du microcontrôleur

Figure 3-2: Principe de fonctionnement du programmeur

NOTE: Pour plus d'informations concernant le programmeur *AVRprog* et *AVRflash*, consultez le manuel *AVRflash*.

Les microcontrôleurs AVR sont programmés au moyen de la communication série SPI utilisant les broches MISO, MOSI et SCK du microcontrôleur.



Pendant la programmation, un multiplexeur déconnecte les broches du microcontrôleur utilisées pour la programmation du reste de la carte, et les connecte au programmeur AVRprog. Une fois la programmation achevée, ces broches sont déconnectées du programmeur et peuvent ainsi être utilisées comme des lignes d'entrées/sorties.

4.0. Programmeur externe AVRISP mkII

En plus du programmeur embarqué, le système de développement EasyAVR6 peut également utiliser le programmeur externe AVRISP de Atmel pour la programmation des microcontrôleurs. Un tel programmeur se connecte au connecteur AVRISP. Afin de configurer la carte pour permettre la programmation du microcontrôleur via le programmeur externe, J8 devra occuper la position EXTERNAL avant l'allumage du programmeur. Utilisez ensuite J7 pour choisir la socket cible.



Figure 4-1: Placement du jumper J7

J8 en position EXTERNAL active la programmation via le programmeur extérieur AVRISP

J8 en position ON-BOARD active la programmation via le programmeur embarqué AVRprog



Position du jumper J7 pour l'utilisation du programmeur extérieur afin de programmer les microcontrôleurs de format DIP20B ou DIP8



Position du jumper J7 pour l'utilisation du programmeur extérieur afin de programmer les microcontrôleurs de format DIP14



Position du jumper J7 pour l'utilisation du programmeur extérieur afin de programmer les microcontrôleurs de format DIP20A ou DIP40



Position du jumper J7 pour l'utilisation du programmeur extérieur afin de programmer les microcontrôleurs de format DIP28

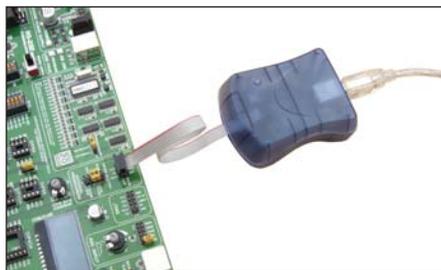


Figure 4-2: AVRISP mkII est connecté au système de développement

5.0. Connecteur JTAG

JTAG ICE est un émulateur utilisé par les microcontrôleurs AVR munis d'interface JTAG intégrés (microcontrôleurs *Mega AVR*). *JTAG ICE* a été initialement conçu pour les utilisateurs du programme *AVR Studio*. Le connecteur *JTAG* intégré aux microcontrôleurs *AVR* est une version modifiée de l'interface *JTAG* original. Il permet de modifier le contenu des mémoires internes *EEPROM* et *FLASH* (programmation du microcontrôleur).



Figure 5-1: Connecteur JTAG

L'émulateur *JTAG ICE* utilise un connecteur mâle 2x5 pour établir la connexion avec le système de développement



Figure 5-2: *JTAGICE mkII* connecté au système de développement

Le connecteur *JTAG* est directement relié aux broches du microcontrôleur. Son fonctionnement ne dépend donc pas des *jumpers* J7 et J8.

6.0. Oscillateur

Un oscillateur est fourni avec la carte. Ce quartz est utilisé pour fournir un signal d'horloge au microcontrôleur. Il est connecté à une interface de laquelle vous pourrez le déconnecter afin de le remplacer par un autre quartz. La fréquence d'horloge ne pourra en aucun cas excéder la fréquence de fonctionnement maximale du microcontrôleur.

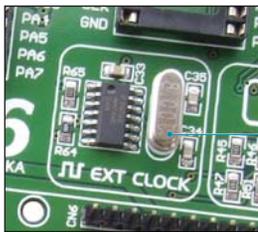


Figure 6-1: Oscillateur

Quartz X2 branché à son interface, ce qui lui permet d'être facilement remplaçable

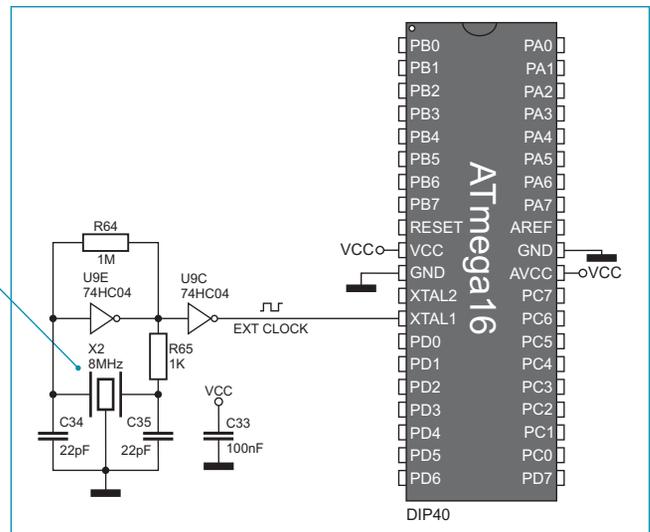


Figure 6-2: Schéma de l'oscillateur

7.0. Alimentation électrique

Le système de développement *EasyAVR6* supporte deux types d'alimentation électrique:

1. Alimentation +5V PC fournit par le câble USB;
2. Alimentation externe reliée au connecteur AC/DC de la carte de développement.

Le régulateur de tension MC34063A et le redresseur de Gretz autorisent une alimentation AC (entre 7V et 23V) et DC (entre 9V et 32V). Le *jumper* J6 est utilisé pour la sélection du type d'alimentation. Lorsque la carte est alimentée par USB, le *jumper* J6 doit être placé en position USB. Dans le cas où l'alimentation externe est utilisée, J6 doit être placé sur la position EXT. Pour mettre sous tension et éteindre le système de développement, utilisez l'interrupteur OFF/ON POWER SUPPLY.

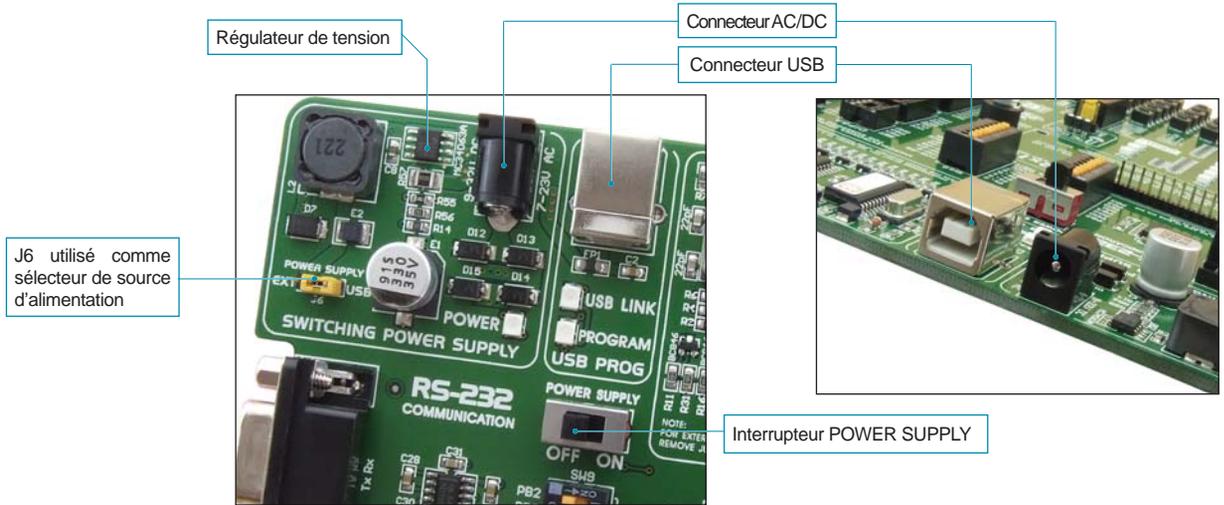


Figure 7-1: Alimentation électrique

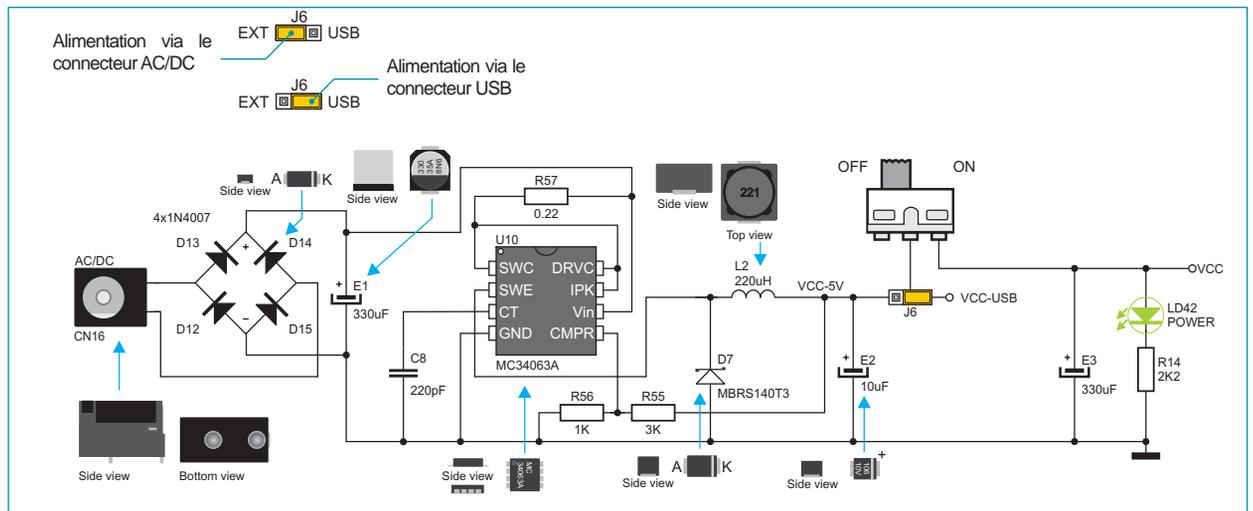


Figure 7-2: Schéma de l'alimentation électrique

page 8.0. Interface de communication RS-232

La communication série RS-232 est réalisée par l'intermédiaire du connecteur 9-broches SUB-D et du module USART du microcontrôleur. Pour rendre possible une telle communication, il faut connecter les lignes de communication RX et TX aux broches du microcontrôleur contrôlant le module USART. Ceci s'effectue par l'intermédiaire du *DIP switch* SW9 (les lignes CTS et RTS étant optionnelles). Les broches du microcontrôleur utilisées pour ce type de communication sont les suivantes: RX - donnée reçue, TX - donnée transmise, CTS - *clear to send* (accusé de réception de la requête) et RTS - *Request To Send* (Requête d'envoi). Le débit atteint 115 kbps. USART (*universal synchronous/asynchronous receiver/transmitter*) est un des moyens les plus fréquemment utilisés pour l'échange de données entre le PC et les périphériques. Pour activer le module USART du microcontrôleur afin de recevoir des signaux d'entrées de tensions différentes, l'utilisation d'un convertisseur de tension tel que le MAX-202C est indispensable.

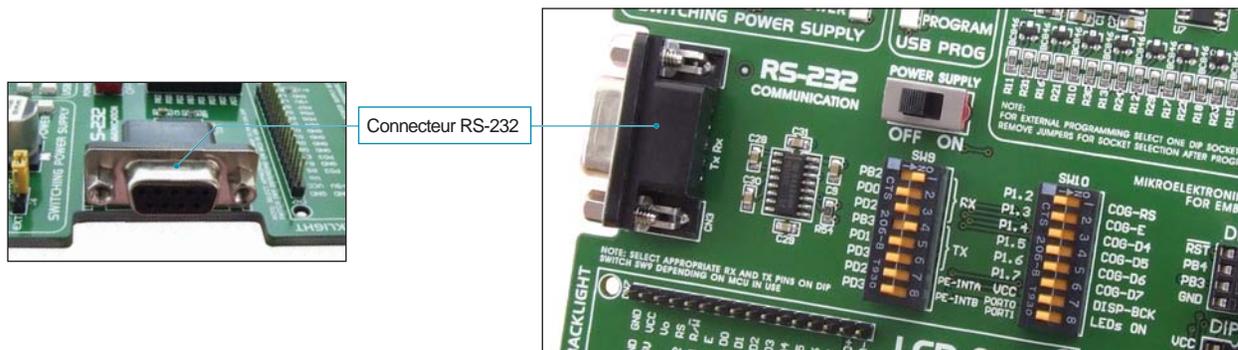


Figure 8-1: Module RS-232

Le *DIP switch* SW9 sert à déterminer quelles broches du microcontrôleur doivent être utilisées en réception et transmission. Les broches de sortie du microcontrôleur dépendent du type de microcontrôleur utilisé. La Figure 8-2 présente un microcontrôleur placé en DIP40 (*ATMEGA16*).

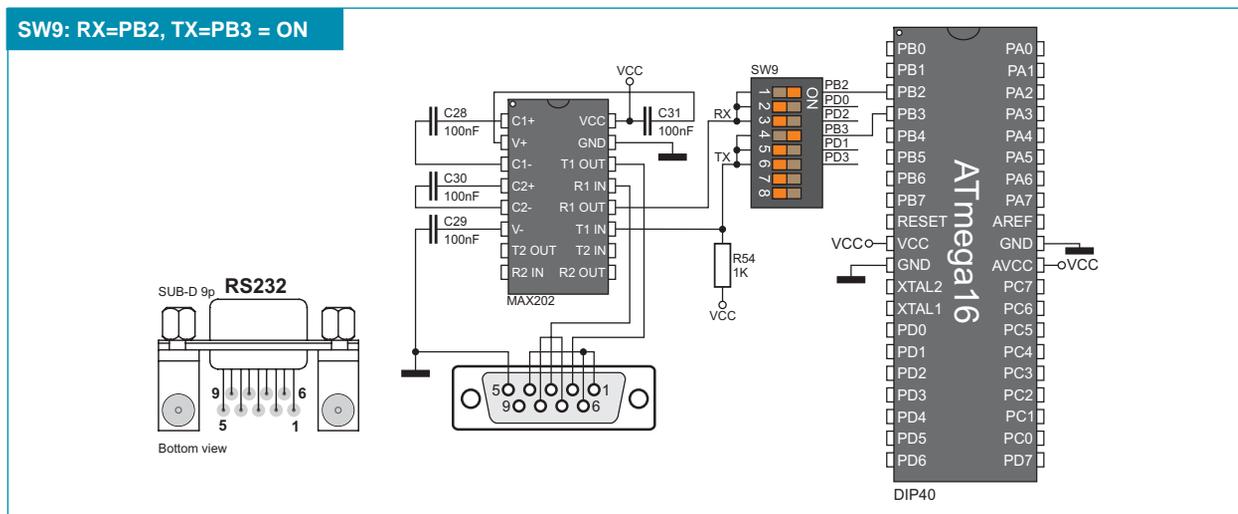


Figure 8-2: Schéma du module RS-232

NOTE: Assurez vous que votre microcontrôleur est fourni avec le module USART. En effet, les microcontrôleurs AVR n'intègrent pas tous ce module.

9.0. Interface de communication PS/2

Un périphérique comme un clavier ou une souris peut être connecté via le connecteur PS/2. L'activation de la communication PS/2 s'effectue en positionnant correctement les *jumpers* J16 et J17, c'est à dire en reliant les lignes DATA et CLK aux broches PC0 et PC1 du microcontrôleur. Ne jamais connecter/déconnecter le périphérique lorsque le système de développement est allumé, car des dommages irréversibles pourraient être infligés au microcontrôleur.

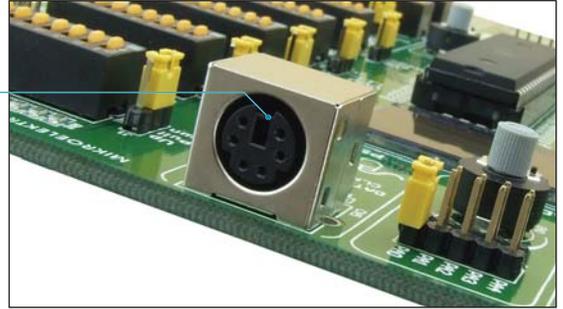


Figure 9-1: Communication PS/2 désactivée (J16 et J17 sont déconnectés)



Figure 9-2: Communication PS/2 activée (J16 et J17 sont placés correctement)

Connecteur PS/2



Jumpers J16 et J17 positionnés

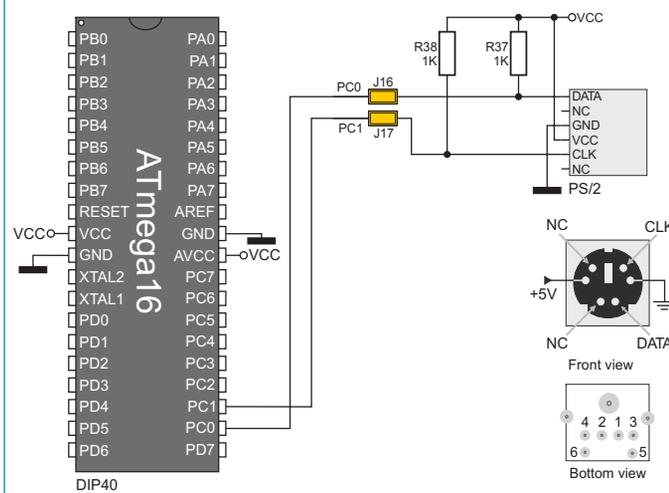


Figure 9-3: Schéma de connexion du connecteur PS/2



Figure 9-4: EasyAVR6 relié à un clavier

page 10.0. Capteur de température DS1820

La communication série 1-wire® permet la transmission de données par l'intermédiaire d'une seule ligne de communication, ce qui signifie qu'une seule broche du microcontrôleur est utilisée. Le processus est sous le contrôle du microcontrôleur (maître). Tous les périphériques esclaves ont par défaut un unique code ID. Ceci permet au maître d'identifier facilement les périphériques esclaves partageant le même interface.

Le capteur de température DS1820 utilise le standard 1-wire. Il est capable de mesurer les températures comprise entre -55 et 125°C avec une marge d'erreur de $\pm 0.5^\circ\text{C}$ pour les températures comprises entre -10 to 85°C. Ce capteur requiert une tension d'alimentation comprise entre 3V et 5.5V. La température est calculée avec une résolution de 9 bits en moins de 750ms. Le système de développement *EasyAVR6* est équipé d'une interface de connexion destinée au DS1820. Le capteur peut utiliser PA4 ou PB2 comme broche de communication avec le microcontrôleur. La sélection de la broche assurant la communication 1-wire avec le microcontrôleur se fait grâce au *jumper* J9. La Figure 10-4 présente une communication 1-wire® avec le microcontrôleur via PA4.



Figure 10-1: la communication 1-wire n'est pas utilisée



Figure 10-2: J9 en position gauche (communication 1-wire via PA4)



Figure 10-3: J9 en position droite (communication 1-wire via PB2)

NOTE: Assurez vous que le demi cercle inscrit sur la carte correspond au côté arrondi du DS1820

Jumper J9 en position PA4

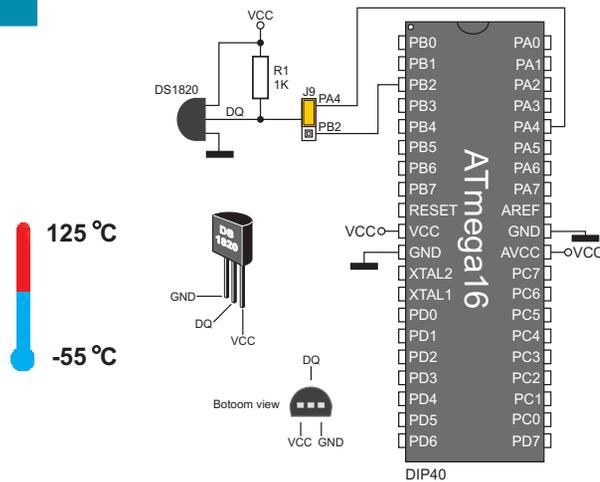


Figure 10-4: Schéma représentant la communication 1-wire

11.0. Convertisseur Analogique/Numérique (ADC)

Un convertisseur Analogique/Numérique a pour rôle de convertir un signal analogique en données numériques. Le convertisseur A/N est linéaire ce qui signifie que la donnée convertie est linéairement dépendante à la tension d'entrée.

Le convertisseur A/N situé dans le microcontrôleur convertit une tension analogique en un nombre codé sur 10 bits. Une tension d'entrée variant entre 0V et 5V DC peut être fournie par les entrées de test A/N. Le *jumper* J12 permet de sélectionner la broche sur laquelle la conversion aura lieu, à savoir PA0, PA1, PA2, PA3 ou PA4. La résistance R63 a une fonction de protection dans la mesure où elle sert à limiter le courant à travers le potentiomètre et les broches du microcontrôleur. La valeur de l'entrée analogique peut être modifiée linéairement en utilisant le potentiomètre P1 (10k).



Figure 11-1: ADC (position du *jumper* par défaut)



Figure 11-2: PA0 est utilisée comme entrée du convertisseur A/N

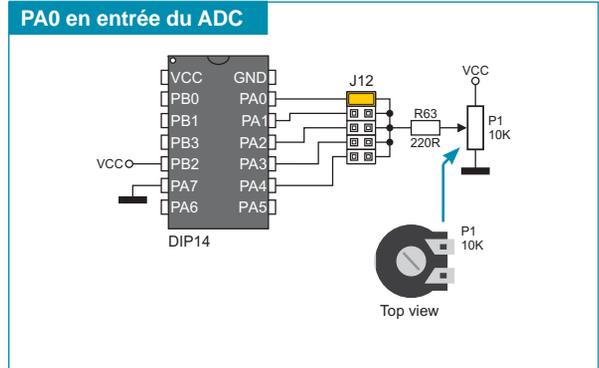


Figure 11-3: Microcontrôleur AVR en DIP14 connecté aux entrées de test du convertisseur A/N

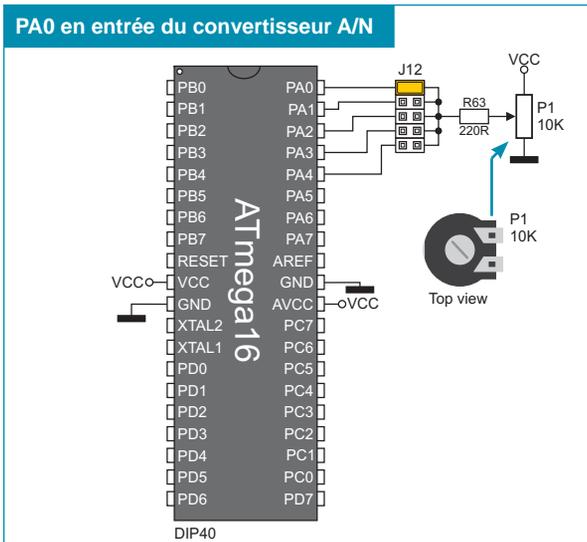


Figure 11-4: Microcontrôleur AVR en DIP40 connecté aux entrées de test du convertisseur A/N

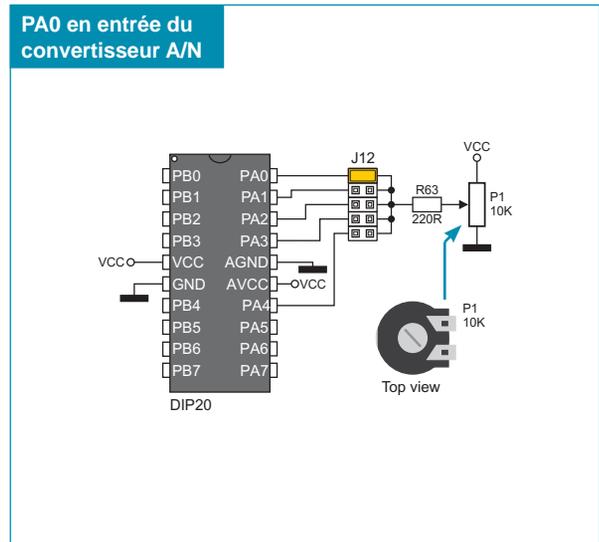


Figure 11-5: Microcontrôleur AVR en DIP20B connecté aux entrées de test du convertisseur A/N

NOTE: Afin d'obtenir une conversion A/N précise, il est nécessaire d'éteindre les LEDs et de déconnecter les *jumpers* des résistances de tirages des broches utilisées par le convertisseur A/N.

12.0. LEDs

Une diode LED (*Light-Emitting Diode*) est un composant électronique particulier utilisé comme source de lumière. Pour chacune des LEDs, l'utilisation d'une résistance pour limiter le courant est indispensable. Sa valeur est déterminée grâce à la formule $R=U/I$ où R correspond à la valeur de la résistance en ohms, U la tension et I le courant traversant la LED. La tension aux bornes d'une LED standard est de l'ordre de 2.5V tandis que le courant peut varier entre 1mA et 20mA. L'intensité du courant qui traverse les LED de *EasyAVR6* est de 1mA.

EasyAVR6 possède 35 LEDs dont le but est d'indiquer visuellement (présence ou non de lumière) l'état de chacune des broches E/S du microcontrôleur. Une LED allumée (active) traduit la présence d'un 1 logique sur la broche. Utilisez le *DIP switch* SW8 afin de choisir les ports (PORTA/E, PORTB, PORTC ou PORTD) que vous souhaitez visualiser.

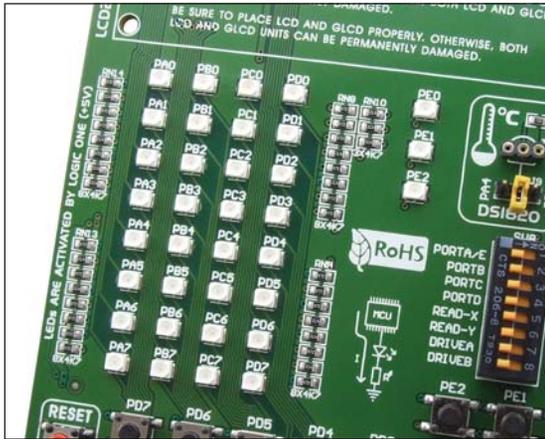
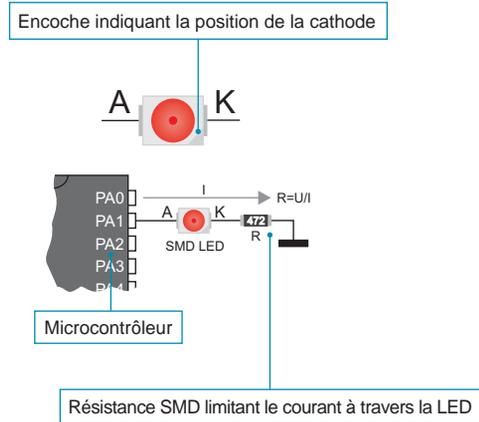


Figure 12-1: LEDs



SW8: PORTA = ON

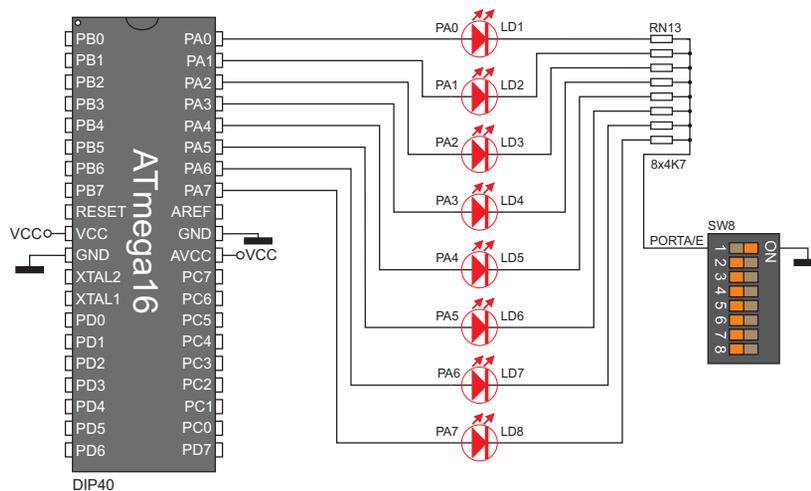


Figure 12-2: Schéma des connexions entre le PORTA et ses diodes respectives

13.0. Boutons poussoirs

L'état logique de toutes les entrées numériques du microcontrôleur peuvent être modifiées par l'intermédiaire des boutons poussoirs. Le *jumper* J13 sert à définir l'état logique qui doit être appliqué à la broche du microcontrôleur lorsque le bouton associé est pressé. Des résistances de protection ont pour but de limiter le courant maximum afin de prévenir d'éventuels court-circuits. Si besoin est, les utilisateurs expérimentés ont la possibilité de court-circuiter ces résistances grâce au *jumper* J18. A coté de ces boutons poussoirs, on trouve le bouton RESET. Celui-ci n'est pas relié directement à la broche MCLR. Le signal *reset* (réinitialisation) est généré par le programmeur.

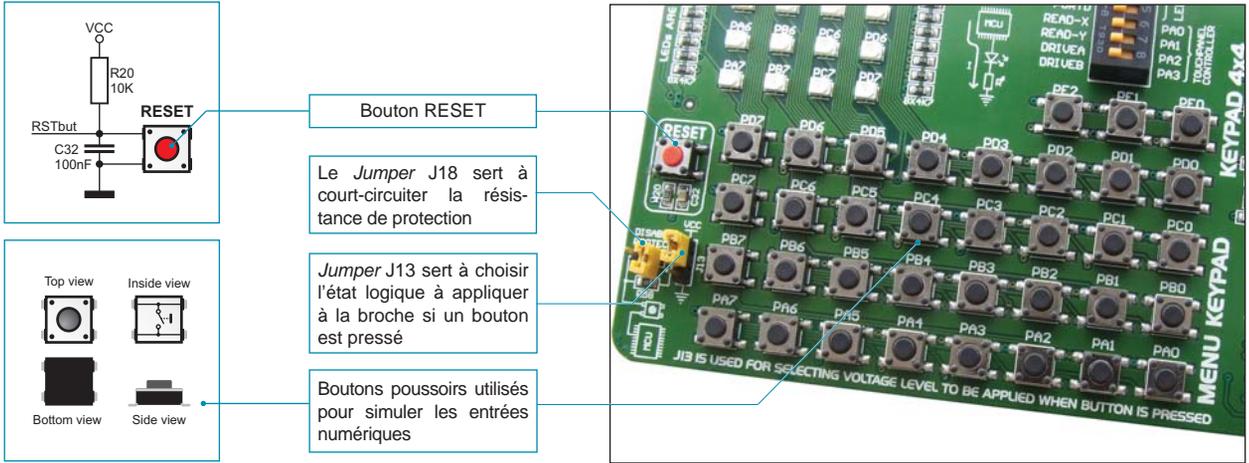


Figure 13-1: Boutons poussoirs

Pressez n'importe quel bouton poussoir (PA0-PA7) lorsque J13 est dans la position VCC, et un 1 logique (5V) sera appliqué à la broche associée du microcontrôleur (voir Figure 13-2).

Jumper J13 en position VCC

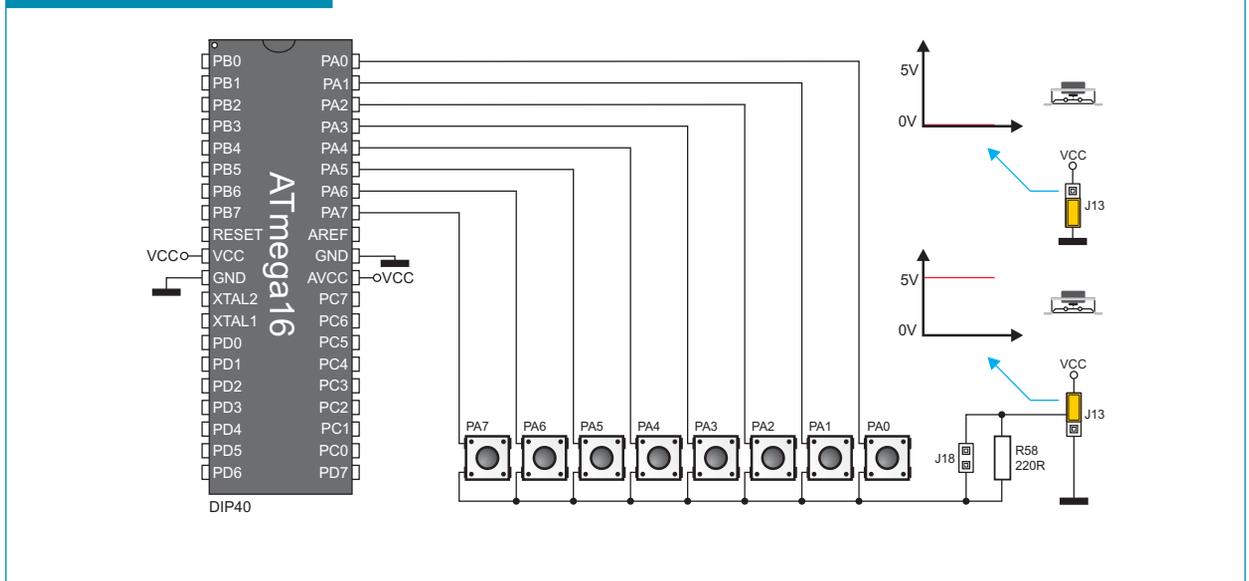


Figure 13-2: Schéma de connexion des boutons poussoirs du PORTA

14.0. Claviers numériques

Le système de développement *EasyAVR6* comporte deux claviers numériques (*keypads*): *keypad 4x4* et *keypad MENU*. *Keypad 4x4* est un clavier alphanumérique standard relié au PORTC du microcontrôleur. Il est basé sur le principe "scan and sense" où PC0, PC1, PC2 et PC3 sont configurées en entrées et connectées aux résistances de *pull-down* (reliées à la masse). Les broches PC4, PC5, PC6 et PC7 sont configurées en sorties de niveau haut (1 logique). La pression d'un des boutons va causer l'application d'un 1 logique sur une des broches d'entrées. La détection du bouton pressé s'effectue par software. Afin de déterminer quel bouton poussoir a été pressé, un 1 logique sera appliqué successivement à chacune des broches de sorties (PC4, PC5, PC6 et PC7). Par exemple, si le bouton '6' est pressé, un 1 logique va apparaître en PC2 lorsqu'un 1 logique sera envoyé sur la sortie PC5.

Les boutons du *Keypad MENU* sont connectés au PORTA de manière similaire. La seule différence est dans l'agencement du clavier numérique. Les boutons du *keypad MENU* sont disposés de manière à faciliter la navigation à travers les menus.



Figure 14-1: *Keypad 4x4*

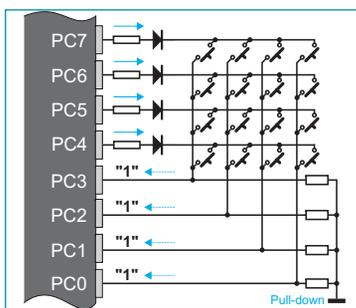


Figure 14-2: Principe du *Keypad 4x4*

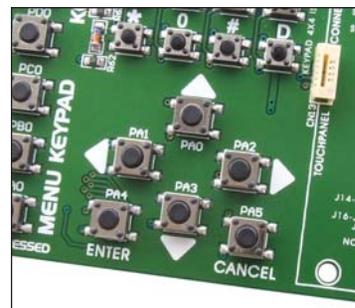


Figure 14-3: *Keypad MENU*

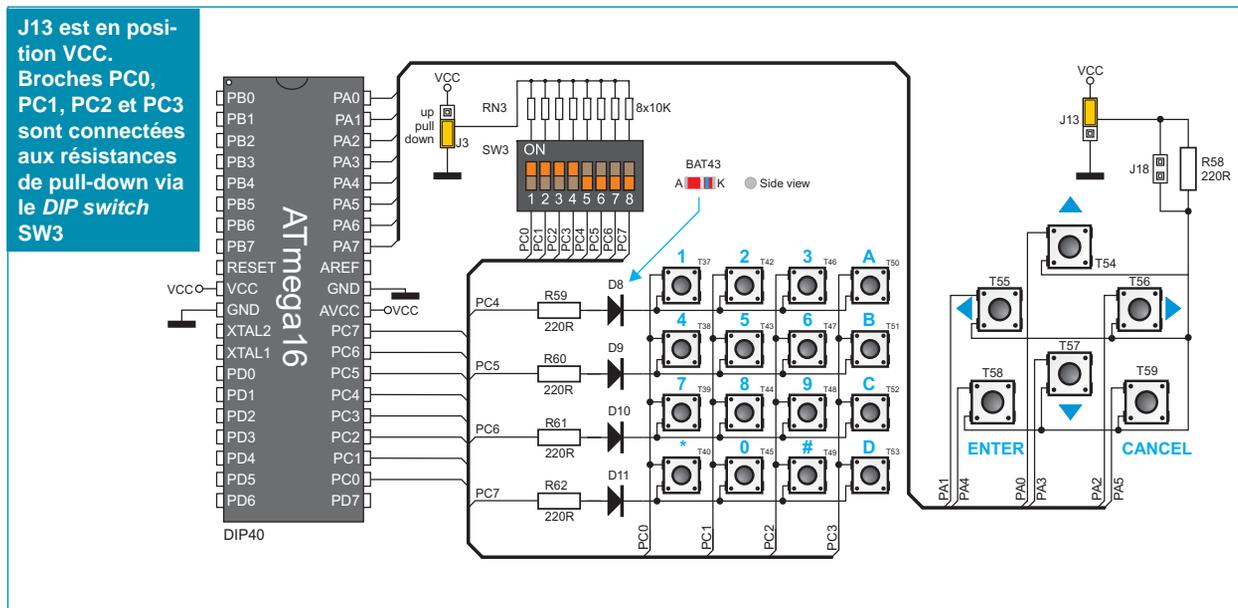


Figure 14-4: Schéma des connexions entre les claviers numériques (*4x4* et *MENU*) et le microcontrôleur

15.0. Afficheur LCD alphanumérique 2x16

La carte de développement *EasyAVR6* est équipée d'un connecteur pour afficheur LCD alphanumérique 2x16. Ce connecteur est relié au microcontrôleur via le PORTD. Le potentiomètre P7 est utilisé pour l'ajustement du contraste. L'interrupteur *DISP-BCK* du *DIP switch* SW10 sert à allumer/éteindre le rétro-éclairage de l'afficheur

La communication entre le microcontrôleur et l'afficheur LCD s'effectue en mode 4 bits (*4-bit mode*). Les symboles alphanumériques sont représentés sur deux lignes, chacune pouvant contenir jusqu'à 16 caractères de taille 7x5 pixels.



Figure 15-1: Connecteur de l'afficheur LCD alphanumérique



Figure 15-2: Afficheur LCD 2x16

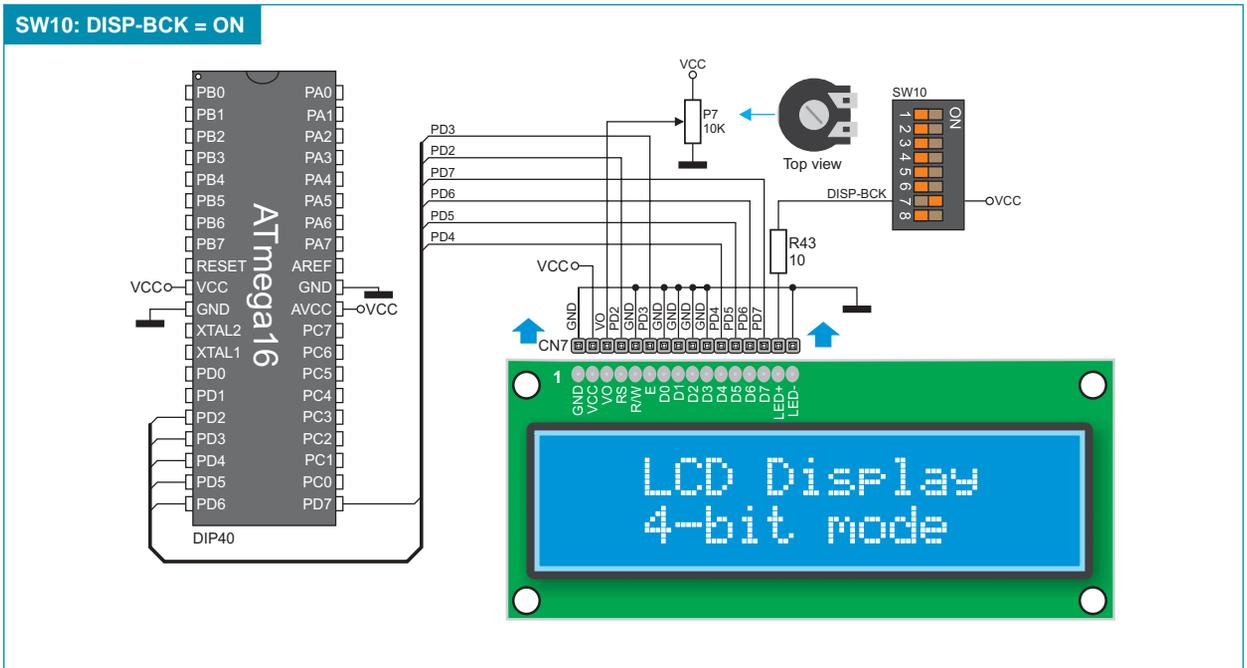


Figure 15-3: Schéma de branchement de l'afficheur LCD 2x16

17.0. Afficheur graphique LCD 128x64

L'afficheur graphique LCD 128x64 (128x64 GLCD) permet l'affichage de données graphique complexes. Il communique avec le microcontrôleur par l'intermédiaire des PORTC et PORTD. L'afficheur GLCD a une résolution de 128x64 pixels, ce qui rend possible l'affichage de diagrammes, tableaux et autres graphiques. Comme le PORTD est également utilisé par l'afficheur LCD alphanumérique 2x16, vous ne pourrez pas vous servir simultanément de ces deux afficheurs. Le potentiomètre P6 sert à ajuster le contraste de l'afficheur GLCD. L'interrupteur 7 du *DIP switch* SW10 est utilisé pour activer/désactiver le rétro-éclairage.



Figure 17-1: Afficheur GLCD

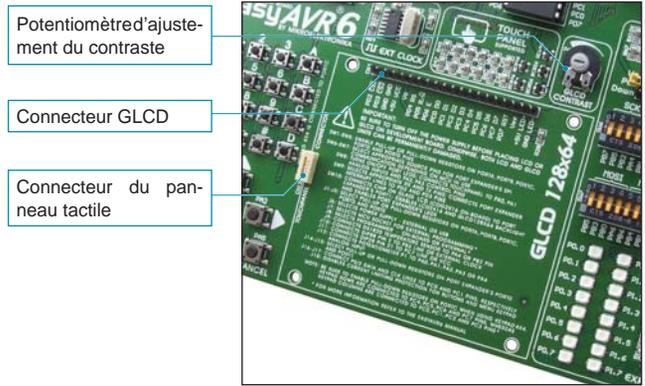


Figure 17-2: Connecteur GLCD

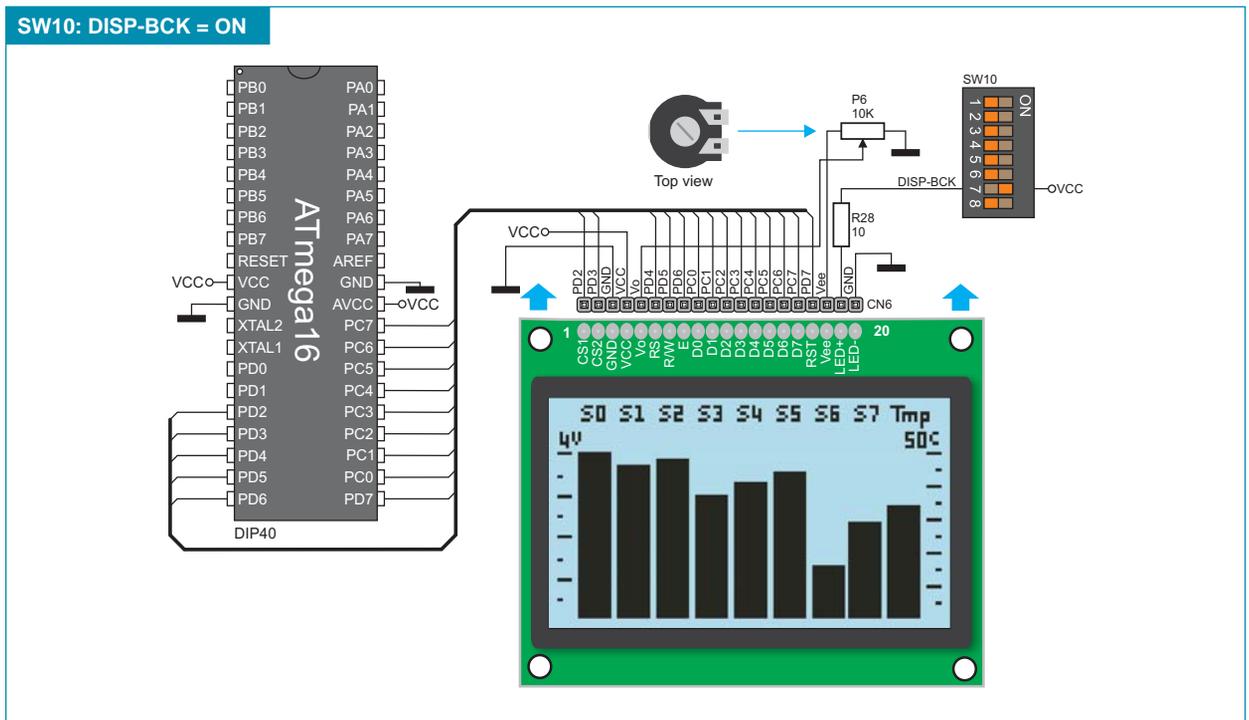


Figure 17-3: Schéma de connexion de l'afficheur GLCD

page 18.0. Panneau tactile (*touch panel*)

Le *touch panel* est un panneau fin, adhésif et transparent sensible au toucher. Il se place sur l'afficheur GLCD et le "transforme" ainsi en écran tactile. Son intérêt réside dans le fait qu'il enregistre les pressions effectuées sur l'afficheur et envoie ses coordonnées sous forme d'un signal analogique au microcontrôleur. Les interrupteurs 5,6,7 et 8 du *DIP switch* SW8 sont utilisés pour relier le *touch panel* au microcontrôleur.

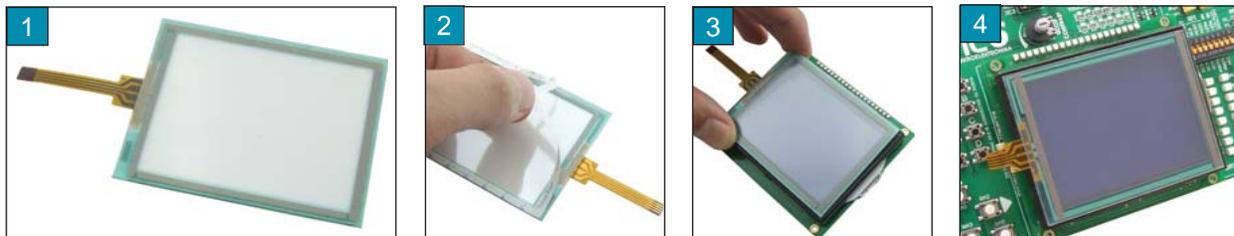


Figure 18-1: panneau tactile

La Figure 18-1 vous montre comment placer le *touch panel* sur l'afficheur GLCD. Assurez vous que le câble plat est situé sur la gauche de l'afficheur GLCD, comme décrit Figure 4.

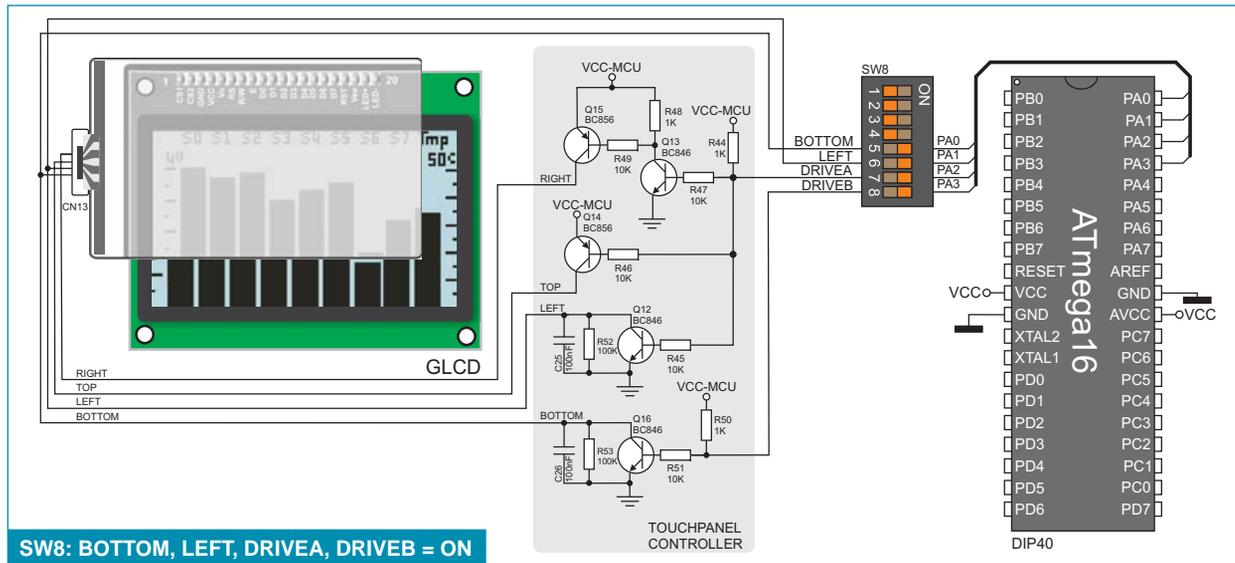


Figure 18-2: Schéma de connexion du *Touch panel*



Figure 18-3: Connexion du *touch panel*

La Figure 18-3 présente en détail comment connecter le *touch panel* au système de développement. Positionnez l'extrémité du câble plat sur le connecteur CN13, (Figure 1). Branchez le câble au connecteur (Figure 2) et presser doucement (Figure 3). Seulement après avoir effectué ces opérations, vous pourrez connecter l'afficheur GLCD au connecteur approprié (Figure 4).

NOTE: Les LEDs et les résistances de tirage *pull-up/pull-down* du PORTA doivent être désactivées pendant l'utilisation du *touch panel*.

19.0. Ports d'Entrées/Sorties

Sur la partie droite de la carte de développement se trouve sept *DIP switches* ayant chacun dix interrupteurs. Chacun d'eux est connecté aux ports d'E/S du microcontrôleur. Certaines des lignes d'E/S sont directement reliées aux broches du microcontrôleur tandis que d'autres sont connectées au microcontrôleur par l'intermédiaire de *jumpers*. Les *DIP switches* SW1-SW5 servent à relier les broches aux résistances de tirage *pull-up/pull-down*. La position des *jumpers* J1-J5 détermine si les ports utilisent une résistance de *pull-up* ou de *pull-down*.

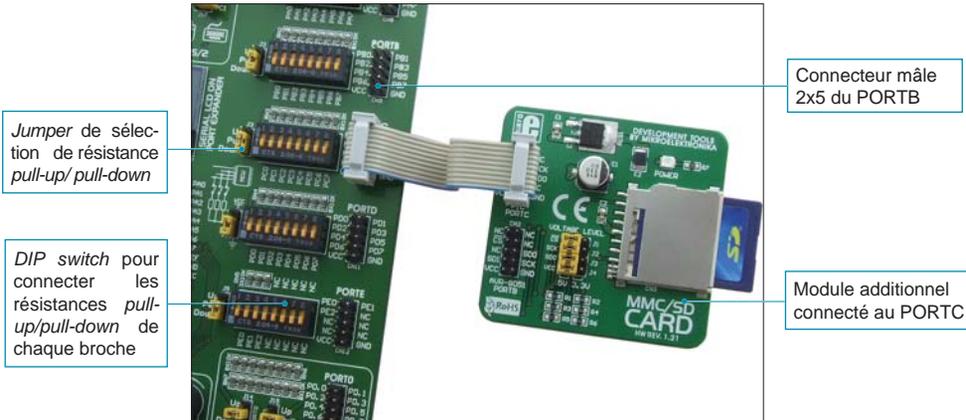


Figure 19-1: Ports E/S

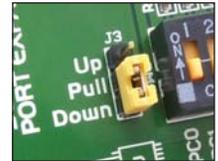


Figure 19-2: J3 en position pull-down

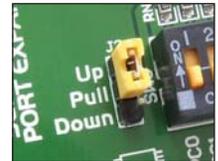


Figure 19-3: J3 en position pull-up

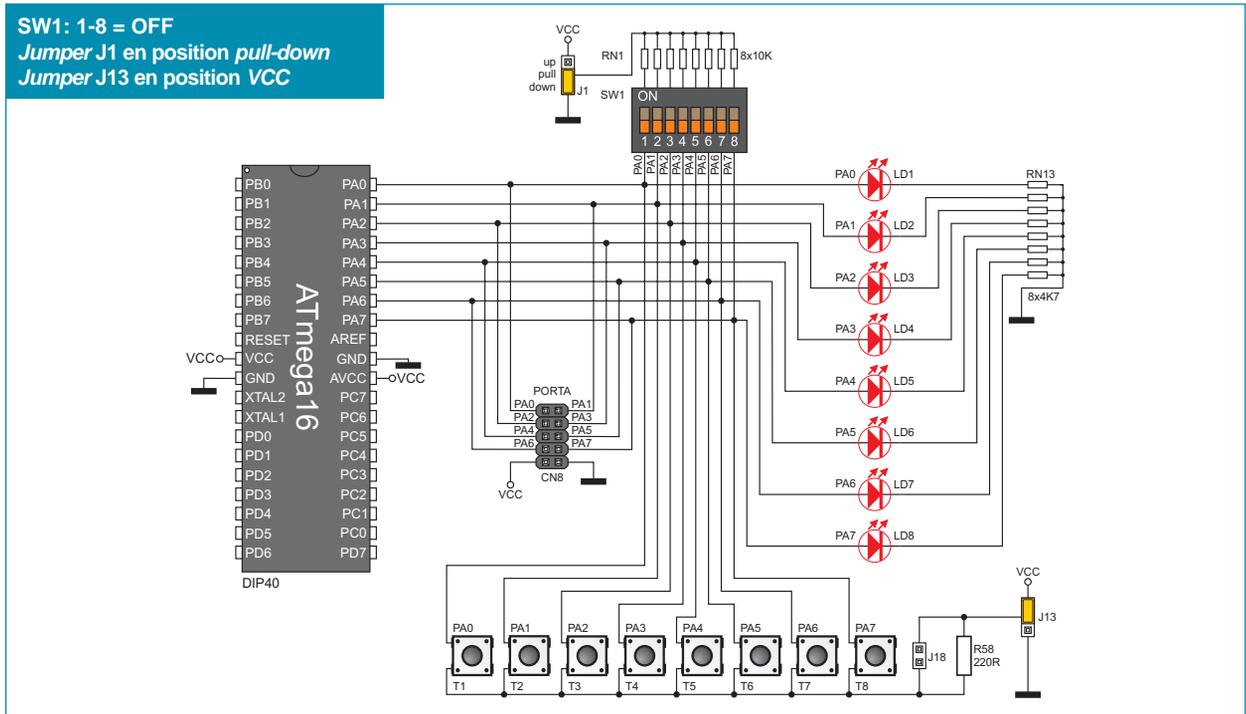


Figure 19-4: Schéma de connexion du PORTA

Les résistances de *Pull-up/pull-down* vous permettent de forcer l'état logique de toutes les entrées du microcontrôleur en mode *idle* ("en attente"). Cet état dépend de la position des *jumpers* J1 et J13. La broche PA0 et son *DIP switch* associé SW1, le *jumper* J1 et le bouton poussoir PA0 avec le *jumper* J13 sont utilisés ici dans le but de vous expliquer le rôle des résistances de tirage. Leur principe de fonctionnement est le même pour toutes les broches du microcontrôleur.

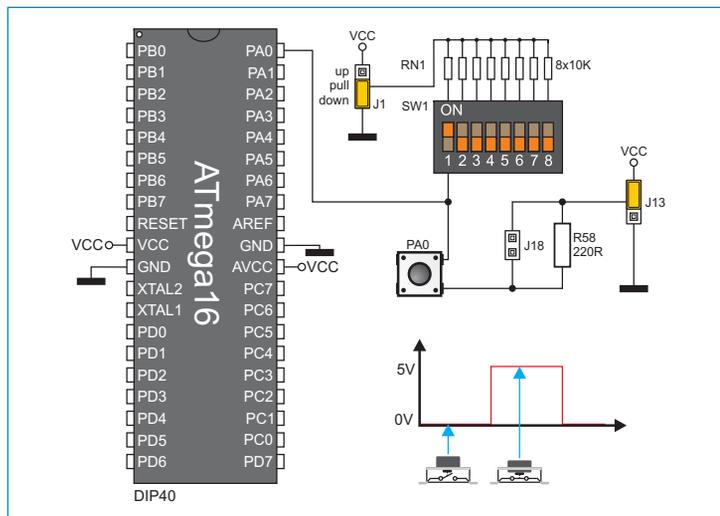


Figure 19-5: *Jumper* J1 en position *pull-down* et J13 en position *pull-up*

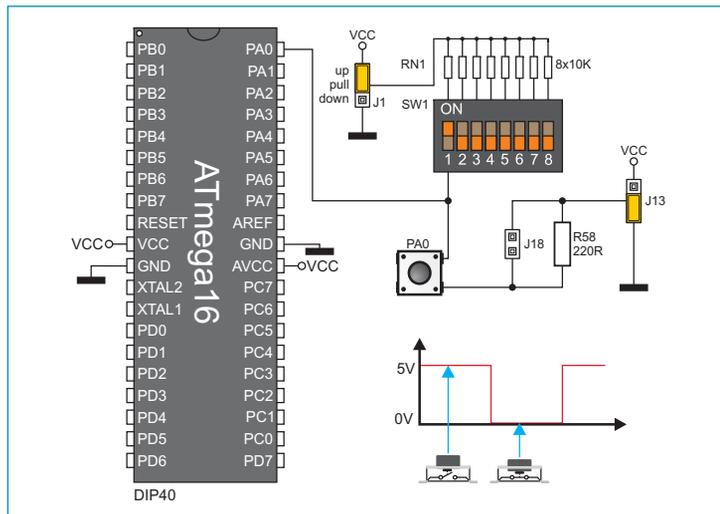


Figure 19-6: *Jumper* J1 en position *pull-up* et J13 en position *pull-down*

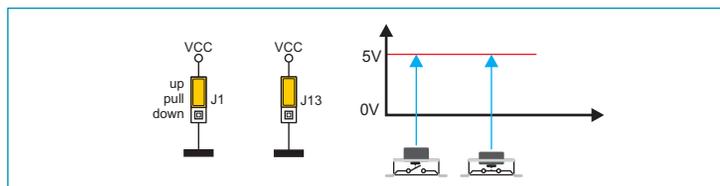


Figure 19-7: *Jumpers* J2 et J13 dans la même position

Afin de permettre la connexion des broches du PORTA aux résistances de *pull-down*, il faut tout d'abord mettre le *jumper* 1 en position basse (à la masse). Ainsi, toutes les broches du PORTA en mode *idle* se verront attribuer un zéro logique (0V) par l'intermédiaire du *jumper* J1 et de la résistance 8x10K. Pour fournir un tel signal à la broche PA0, il est indispensable de placer l'interrupteur de la broche PA0 du *DIP switch* SW1 sur la position ON. Placez le *jumper* J13 en position VCC (*pull-up*).

En conséquence, chaque fois que vous presserez le bouton poussoir PA0, un 1 logique apparaîtra sur la broche PA0. Ce 1 logique provient du fait que J13 est en position VCC.

Afin de permettre la connexion des broches du PORTA aux résistances de *pull-up*, il faut tout d'abord mettre le *jumper* 1 en position haute (VCC). Ainsi, toutes les broches du PORTA se verront attribuer un 1 logique (5V) en mode *idle* par l'intermédiaire du *jumper* J1 et de la résistance 8x10K. Pour fournir un tel signal à la broche PA0, il est indispensable de placer l'interrupteur de la broche PA0 du *DIP switch* SW1 sur la position ON. Placez le *jumper* J13 en position basse (*pull-down*).

En conséquence, chaque fois que vous presserez le bouton poussoir PA0, un zéro logique (0) apparaîtra sur la broche PA0. Ce 0 logique provient du fait que le *jumper* J13 est placé en position GND (à la masse).

Dans le cas où J1 et J13 occupent la même position, la pression d'un bouton poussoir n'entraînera aucune modification de l'état logique sur les broches d'entrées du microcontrôleur.

20.0. Extenseur de port (ports E/S additionnels)

Les lignes de communications SPI et le circuit MCP23S17 vous donnent la possibilité d'augmenter de deux le nombre de ports d'E/S disponibles. Si l'extenseur de port est connecté (par l'intermédiaire des *DIP switch* SW6 et SW7), les broches du microcontrôleurs utilisées pour la communication SPI ne pourront pas être utilisées en tant que broches E/S. Les interrupteurs INTA et INTB du *DIP switch* SW9 activent les interruptions.

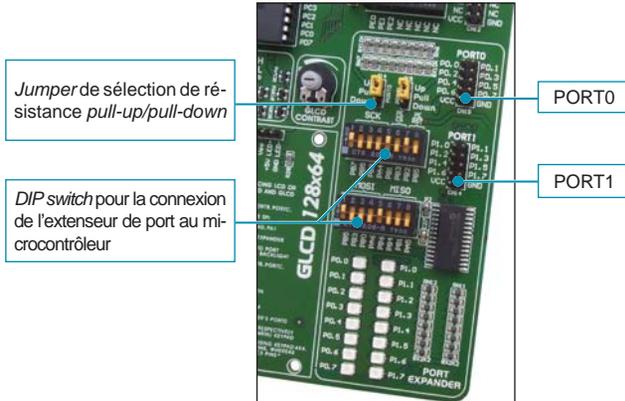


Figure 20-1: Extenseur de port

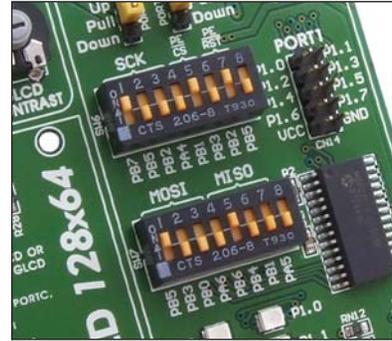


Figure 20-2: DIP switch SW6 et SW7 lorsque l'extenseur de port est activé

Le microcontrôleur communique avec l'extenseur de port (circuit MCP23S17) en utilisant la communication série (SPI). Ce type de communication présente l'avantage de n'utiliser que cinq lignes et d'être full-duplex:

- MOSI - *Master Output, Slave Input* (microcontrôleur en sortie, MCP23S17 en entrée)
- MISO - *Master Input, Slave Output* (microcontrôleur en entrée, MCP23S17 en sortie)
- SCK - *Serial Clock* (signal d'horloge venant du microcontrôleur)
- CS - *Chip Select* (active le transfert de données)
- RST - *Reset* (réinitialisation)

Le transfert de données s'effectue simultanément dans les deux directions (lignes MOSI et MISO). La ligne MOSI est utilisée pour le transfert de données du microcontrôleur vers l'extenseur de port, tandis que la ligne MISO sert au transfert de données de l'extenseur de port au microcontrôleur. Le microcontrôleur commence le transfert de données lorsque CS est à l'état bas (0V). Pour cela, il envoie un signal d'horloge (SCK) qui permet de débiter l'échange de données.

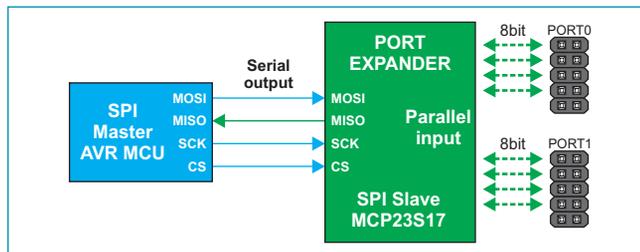


Figure 20-3: Diagramme bloc de la communication SPI

Le principe de fonctionnement des ports 0 et 1 diffère peu de celui des autres ports du système. La seule différence est que les signaux des ports sont reçus en formats parallèles. Le MCP23S17 se charge donc de convertir ces signaux en format série avant de les envoyer au microcontrôleur. Ainsi, le nombre de lignes nécessaires à la communication entre le microcontrôleur et les ports 0 et 1 est réduit.

SW6: CS#=PB1, RST=PB2, SCK = PB7
 SW7: PB6 =MISO, PB5=MOSI
 J14 et J15 en position pull-up

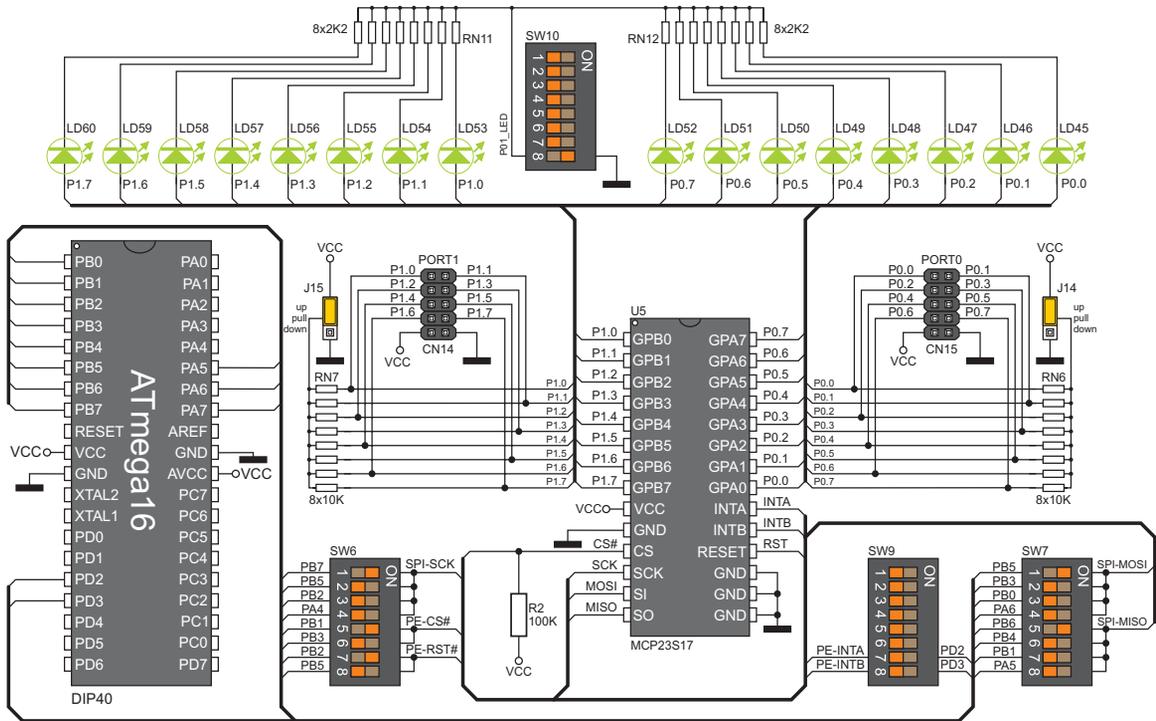


Figure 20-4: Schéma de connexion de l'extenseur de port

COPYRIGHT ET LIMITES DE RESPONSABILITE:

Cet environnement de développement et tous ses documents sont détenus par MikroElektronika. Ces derniers sont protégés par la réglementation sur le copyright ainsi que par les traités internationaux du copyright. Ce manuel est également soumis aux lois sur le copyright. Il ne devra en aucun cas être reproduit dans son intégralité ou de façon partielle (de quelque façon que ce soit) sans l'autorisation préalable écrite de MikroElektronika. L'édition PDF du manuel pourra être imprimée pour un usage privé ou une utilisation locale mais en aucun cas dans le cadre d'une distribution. La modification de ce manuel est interdite.

MikroElektronika fournit ce manuel "tel quel", sans garantie d'aucune sorte, explicite ou implicite, y compris, mais sans limitation les garanties implicites ou les conditions marchandes ou d'adéquation pour un usage particulier.

En aucun cas, MikroElektronika, ses administrateurs, dirigeants, employés ou distributeurs ne pourront être tenus responsables pour tous dommages indirects, spécifiques, accessoires ou consécutifs que ce soient (y compris les dommages pour perte de bénéfice commercial, interruption d'exploitation commerciale, perte d'informations et de données à caractère commercial ou tout autre perte financière) résultant de l'utilisation ou de l'incapacité à pouvoir utiliser les produits MikroElektronika (compilateurs et kits d'évaluation) ou de tout défaut ou erreur dans ce manuel, même si MikroElektronika a été informé de la possibilité de tels dommages.

Les produits et les noms de sociétés apparaissant dans ce manuel peuvent être (ou ne pas être) des marques déposées ou des copyrights de leurs sociétés respectives, Leurs utilisations dans ce manuel ne relève que d'un besoin d'identification ou d'explication à l'avantage du propriétaire et sans intention de nuire.

ACTIVITES A HAUTS RISQUES

Les produits MikroElektronika ne sont ni fault-tolerant (résistant aux pannes) ni conçus, fabriqués ou destinés à être utilisés ou revendus comme: équipements de contrôle de chaînes de production ou d'assemblage dans des conditions dangereuses nécessitant des performances fail-safe (sécurité intégrée), matériels opérant sur des sites nucléaires, systèmes de navigation et de communication d'avions, équipements de contrôle du trafic aérien, matériels médicaux d'assistance à la vie ou systèmes d'armement pour lesquels une défaillance logicielle pourrait entraîner la mort, des blessures ou d'importants dégâts physiques ou environnementaux ("Activités à haut risque"). MikroElektronika et ses fournisseurs déclinent toute responsabilité impliquant la garantie pour toute utilisation dans le cadre d'activités à hauts risques.



MikroElektronika
SOFTWARE AND HARDWARE SOLUTIONS FOR EMBEDDED WORLD ...making it simple

Pour plus d'information sur nos produits, n'hésitez pas à consulter notre site internet: www.mikroe.com

Si vous rencontrez des problèmes avec un de nos produits ou si vous souhaitez obtenir des informations supplémentaires, contactez notre service technique: www.mikroe.com/en/support

Pour toute question, remarque ou proposition commerciale, n'hésitez pas à nous contacter: office@mikroe.com